

**IN THE UNITED STATES DISTRICT COURT
FOR THE SOUTHERN DISTRICT OF WEST VIRGINIA**

HUNTINGTON DIVISION

CHARLES JOHNSON, et al.,

Plaintiffs,

v.

Case No.: 3:13-cv-06529

FORD MOTOR COMPANY,

Defendant.

**MEMORANDUM OPINION and ORDER
REGARDING SOURCE CODE DISCOVERY**

This action involves claims of sudden unintended acceleration in Ford Motor Company (“Ford”) vehicles manufactured between 2002 and 2010, allegedly arising from defects in the electronic throttle control system (“ETC”). The parties have engaged in extensive discovery, including the production by Ford of source code related to the ETC in several vehicle models. However, the parties disagree about how the source code should be examined, used, and tested, and despite numerous attempts by the parties to resolve their disagreements, issues still remain unsettled.

Given the complexities associated with discovery of ETC source code, the court determined that the assistance of an independent technical advisor experienced in computer engineering would hasten the just adjudication of the discovery process “without dislodging the delicate balance of the juristic role.” *Reilly v. United States*, 863 F.2d 149, 156 (1st Cir. 1988). Therefore, exercising its inherent authority, the court appointed William H. Sanders, Ph.D., M.S.E., B.S.E. (“Dr. Sanders”), as a technical advisor for the purpose of assisting with issues related to source code and source code

discovery. *See TechSearch, LLC v. Intel Corp.*, 286 F.3d 1360, 1377 (Fed. Cir. 2002).

The parties subsequently met and conferred with respect to an agenda of discovery disagreements to present to Dr. Sanders. The parties could not agree on all aspects of the agenda; therefore, each party submitted a proposed agenda to the court. After considering the parties' submissions and reviewing the relevant hearing transcripts and orders, the court set an agenda to guide the discussions with Dr. Sanders. (ECF No. 744). Thereafter, the parties and their experts met in person with Dr. Sanders at the secured source code review room in Dearborn, Michigan; they submitted relevant written materials to be examined by Dr. Sanders; and they had follow-up conversations with him. Upon completing an analysis of the outstanding issues and the positions of the parties, Dr. Sanders issued a written report, which set forth his recommendations regarding the code, features, and drivers produced by Ford, as well as Plaintiffs' request for additional tools. (ECF No. 793). The parties were then permitted to submit written responses and objections to Dr. Sanders's report. (ECF Nos. 794, 795). On Thursday, August 25, 2016, the parties appeared, by counsel, with their experts at a hearing before the undersigned and Dr. Sanders to argue their positions with respect to the outstanding source code discovery.

Having now fully considered the issues, the court finds that Dr. Sanders's recommendations should be adopted in their entirety. With regard to the issue of whether additional source code should be produced by Ford, the court finds that Ford should produce certain pieces of ETC-related code not previously provided, which are described herein. However, with respect to non-ETC code (code running in the same binary but not directly involved in ETC functionality), the court does not find that Plaintiffs have demonstrated a genuine need for additional source code, which

outweighs the anticipated burden and expense to Ford. Plaintiffs have yet to adequately test the source code already produced; in part, due to a misunderstanding or disagreement regarding self-created programs and test files. Nevertheless, Plaintiffs have in their possession source code that should allow them the opportunity to reasonably investigate their claims. Therefore, the court **ORDERS** as follows:

1. Source Code, Features, Drivers

The court previously ordered Ford to produce the source code that, when translated into object code, executes the ETC in certain vehicle models, which models were jointly selected by the parties. In addition, Ford was ordered to produce the source code related to watchdogs and fail-safes that insured proper operation of the ETC in said models. Ford produced source code, but Plaintiffs assert that they have been unable to confirm that the source code supplied to them accurately encompasses the ETC and its watchdogs and fail-safes.

First, Plaintiffs complain that they have not received the source code relative to the throttle place position controller (“TPPC”), which should have been produced in response to an earlier order of the court. Ford agrees that Plaintiffs have not received this source code, but claims that it has no ability to produce the code, because the code belongs to a contractor, Visteon Corporation. At the hearing, Ford further advised the court that Visteon Corporation apparently has misplaced/lost the TPPC source code for the vehicles currently being examined by Plaintiffs.

In his report, Dr. Sanders confirms that the TPPC source code should be provided to Plaintiffs, as it provides functionality for one of the key parts of the ETC. Dr. Sanders also explained that the TPPC functionality is located in a separate microprocessor from the ETC functionality in the vehicle models currently being

examined by Plaintiffs. However, some vehicles manufactured by Ford in 2008 and later have the TPPC functionality on the “main micro,” which runs other parts of the ETC code. Ford refers to this TPPC functionality as “internal” versus “external.” Ford acknowledges that the “internal” TPPC code is available, despite the apparent loss of the “external” code. Dr. Sanders recommends that Plaintiffs be provided with ETC and TPPC source code for a vehicle model in which the ETC functionality and TPPC functionality are on the main micro. The undersigned agrees with Dr. Sanders. Accordingly, within **twenty (20) days**, Ford shall provide Plaintiffs with the source code for the ETC, TPPC, and ETC watchdogs and fail-safes for one vehicle model to be selected by Plaintiffs in which the TPPC functionality is “internal.” In regard to the external TPPC source code, Plaintiffs shall promptly issue a subpoena to Visteon Corporation requesting that source code.

Next, Plaintiffs seek additional source code not considered primary to ETC functionality. For example, Plaintiffs request access to files that create the global variables that are used by the ETC code files; to files running in the same task as the ETC code; and to files that control the operation and timing of the tasks sharing the main micro (the powertrain control module). According to Plaintiffs, these files are necessary to determine whether interaction defects can cause unintended acceleration.

In response, Ford argues that Plaintiffs should not be given this additional source code for two reasons. First, the source code already produced to Plaintiffs allows them to fully test the fault tolerance of the ETC. Consequently, additional code is unnecessary to achieve the goals set out by the court in prior orders. Second, when taking into consideration the proportionality requirements that govern the scope of discovery, requiring the production of this additional code would violate Fed. R. Civ.

P. 26. According to Tom Erickson, Ford's Supervisor of Global Software Tools and Processes, Powertrain Engineering, if Ford were required to supply the entire source code for the powertrain control module system, Ford would be forced to spend approximately 150 man hours per vehicle model and model year. (ECF No. 665-1). Ford contends that once engine configurations are added to the equation, source code for 130 different vehicles would have to be produced. Ford argues that such a burden clearly outweighs any anticipated benefit to Plaintiffs; particularly, as Plaintiffs have not conducted a meaningful review of the source code that has already been provided.

In his report, Dr. Sanders concludes that the files creating global variables used by the ETC code are not needed to test the ETC functionality, including its fault tolerance, so long as Plaintiffs are permitted to write test harnesses that can alter the global variables. On the other hand, Dr. Sanders points out that to test for interaction faults, in addition to ETC functionality, it would be useful for Plaintiffs to have the files for the global variables; the files running in the same task as the ETC code; and the files that control the operation and timing of the tasks. Dr. Sanders indicates that the decision regarding whether to order more source code depends upon the nature of Plaintiffs' claims. If the "hypothesis is that the defect is in source code that provides ETC functionality, this hypothesis could be tested using the ETC code alone ... if the hypothesis is that the defect is caused by an interaction between non-ETC and ETC code (e.g. an unintended timing interaction), ... then it would be useful to have access to all source code that runs on the Main Micro to test for the defect." (ECF No. 793 at 6).

The argument over how much source code Ford should produce has persisted for well over a year and has been revisited multiple times by the court. On November

13, 2015, during one of many discussions between the court and counsel concerning source code, the parameters of source code discovery were set as follows:

[Plaintiffs] should be allowed to look at the source code of whatever is in the PCM that directly affects the opening and closing of the throttle plate ... [T]hey should be able to compile that piece of the source code from the PCM and run it in some way so that they can see how it acts and reacts. [T]hey ought to be able to inject faults so they can see what the system does when faults are injected. And ... they ought to be able to see how the watchdogs work because that's Ford's, one of Ford's defenses, is that they've got these redundant watchdogs that will prevent there from being a throttle open more than what the person wants by pressing on the pedal.

(ECF No. 677 at 10-11). These parameters were selected by the court, because Plaintiffs' theory has always been that Ford's ETC system is not fault tolerant, regardless of the nature of the faults. Therefore, identifying specific faults (such as interaction faults) should not be the focus of discovery. Discovery in this case should center on how the ETC functionality—including its watchdogs and fail-safes—works when simultaneously confronted with multiple faults. Taking into account the proportionality argument raised by Ford, as well as Dr. Sanders's findings and recommendations, the undersigned finds that Plaintiffs have sufficient source code already available to them to test their theory. Accordingly, Ford shall not be required to produce additional source code, other than the following: (1) TPPC source code as previously discussed; (2) source code not already produced related to ETC functionality, watchdogs, and fail-safes that ensures that the data written to memory is not corrupted, and that does checksums and error detection and correction functionality in the independent plausibility checker ("IPC"); (3) the PC-Lint configuration files; and (4) files Ford already agreed to produce.

Therefore, to the extent Ford has not produced all of the source code described

in category two above, it shall produce same within **twenty (20) days**. In addition, Plaintiffs asked for PC-Lint configuration files used by the eQuizzer build tool. Dr. Sanders recommends that Plaintiffs be provided with the PC-Lint configuration files, and the court agrees that providing Plaintiffs with these files will improve the efficiency of review. Accordingly, Ford shall produce the PC-Lint configuration files within **twenty (20) days**. Finally, Ford also agreed to produce calibration files for the eQuizzer, and those files shall also be produced within **twenty (20) days**.

Although Ford need not produce source code for other files, Plaintiffs must be permitted to create test harnesses that will allow them to fully evaluate the ETC source code and functionality. Tools needed to accomplish that testing are addressed below.

2. Source Code Tools

Ford and Plaintiffs have worked together to assemble certain software tools for Plaintiffs to use in their analysis and testing of the source code. Plaintiffs want to add the following tools:

- i. Python;
- ii. Cygwin;
- iii. Wiki software; and
- iv. Mathworks Polyspace

Ford objects to the first three tools, but agrees that Plaintiffs may use Mathworks Polyspace.

Dr. Sanders recommends that Plaintiffs be permitted to use Python and Mathworks Polyspace, and to a lesser degree, Cygwin. Ford does not expressly object to these recommendations in its response, and the court finds that these tools are appropriate for the testing to be performed by Plaintiffs. As indicated by Dr. Sanders,

Python shall be limited to the language and libraries provided in the standard distribution of Python, unless at some future date, certain add-ons are required. In that event, Plaintiffs shall meet and confer with Ford to agree on the add-ons to be used. In addition, Plaintiffs shall, to the extent available, use a precompiled version of Python. In regard to Cygwin, Plaintiffs shall first attempt to compile the eQuizzer using the Cosmic compiler. If Plaintiffs determine that Cygwin is still needed, the minimum set of commands within Cygwin necessary to compile the eQuizzer using the provided makefile should be used. Plaintiffs shall not use Cygwin for any purpose other than execution of the eQuizzer makefile.

One issue that arose during the discussions with Dr. Sanders was what restrictions, if any, are or should be placed on Plaintiffs in using programs or object files to build test harnesses. Ford does not object to Plaintiffs using compilers, assemblers, linkers, and associated tools to create programs for automated testing of source code. However, Ford believes that it should be permitted to “review” any such programs in advance to ensure that their functionality is limited to automated testing. Ford’s concern is that Plaintiffs will create programs that modify Ford’s source code and then execute tests using the altered code in violation of an agreed Protective Order Governing Source Code. (ECF No. 630). At the hearing, Plaintiffs advised the court and Ford that they did not intend to create programs for use in the test room that would modify the source code. Plaintiffs stated, however, that they would need the ability to “unlock” source code in actual vehicles and in the Green Hills compiler to perform testing.

Dr. Sanders recommends that Plaintiffs be allowed to use “compilers, assemblers, linkers, and associated tools provided to create programs and object files,

that when linked with Ford-produced source code, can automate testing of the code, for various global variables and other configuration settings.” (ECF No. 793 at 10-11). He further recommends that these programs and object files not be considered “tools” requiring a ten-day notice to Ford prior to usage as set forth in the court’s June 12, 2015 order. (ECF No. 543 at 18). The court agrees with these recommendations. When drafting the June 2015 order, the undersigned never intended to include as “tools” programs and object files created by Plaintiffs to automate testing of the source code. Forcing Plaintiffs to check with Ford every time such a program or file is created or modified would unnecessarily delay the testing process. Plaintiffs should be allowed to conduct robust testing on the source code in a fluid environment without having to suspend its process to obtain approval from Ford. With respect to the issue of whether the Protective Order requires modification to allow Plaintiffs to automate testing or unlock source code, the undersigned finds that neither of these actions results in a change or alteration of the source code in the manner envisioned by the Protective Order. Accordingly, Plaintiffs may proceed with this testing—including injecting faults, changing global variables, and altering configuration settings—without violating the Protective Order.

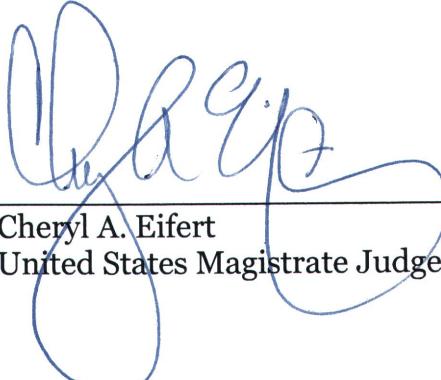
As to Plaintiffs’ request to use Wiki software, Dr. Sanders recommends that Wiki not be permitted due to its cost, complexity, and potential security concerns. Dr. Sanders also mentions that Wiki comes in many different packages, and the specific package proposed by Plaintiffs is unknown. In response, Plaintiffs argue that Wiki poses no security threat to Ford given the physical and electronic isolation of the source code review room. Plaintiffs assert that Wiki would allow them to generate reports more quickly and efficiently as several individuals could work simultaneously on each

report. Plaintiffs did not explicitly provide information regarding the Wiki package they seek to use, or the hardware and/or software support needed for Wiki. In order for the court to evaluate the propriety of Wiki, Plaintiffs shall provide **details** regarding the software version or package of Wiki they wish to use, and the software/hardware support they believe is necessary in order to run that Wiki package.

Plaintiffs are instructed to begin testing of the source code in a manner consistent with this Order. Plaintiffs shall be prepared to provide the court with a status report and estimated time table for completion of source code discovery at the next status conference scheduled in October.

The Clerk is directed to provide a copy of this Order to counsel of record.

ENTERED: September 1, 2016


Cheryl A. Eifert
United States Magistrate Judge